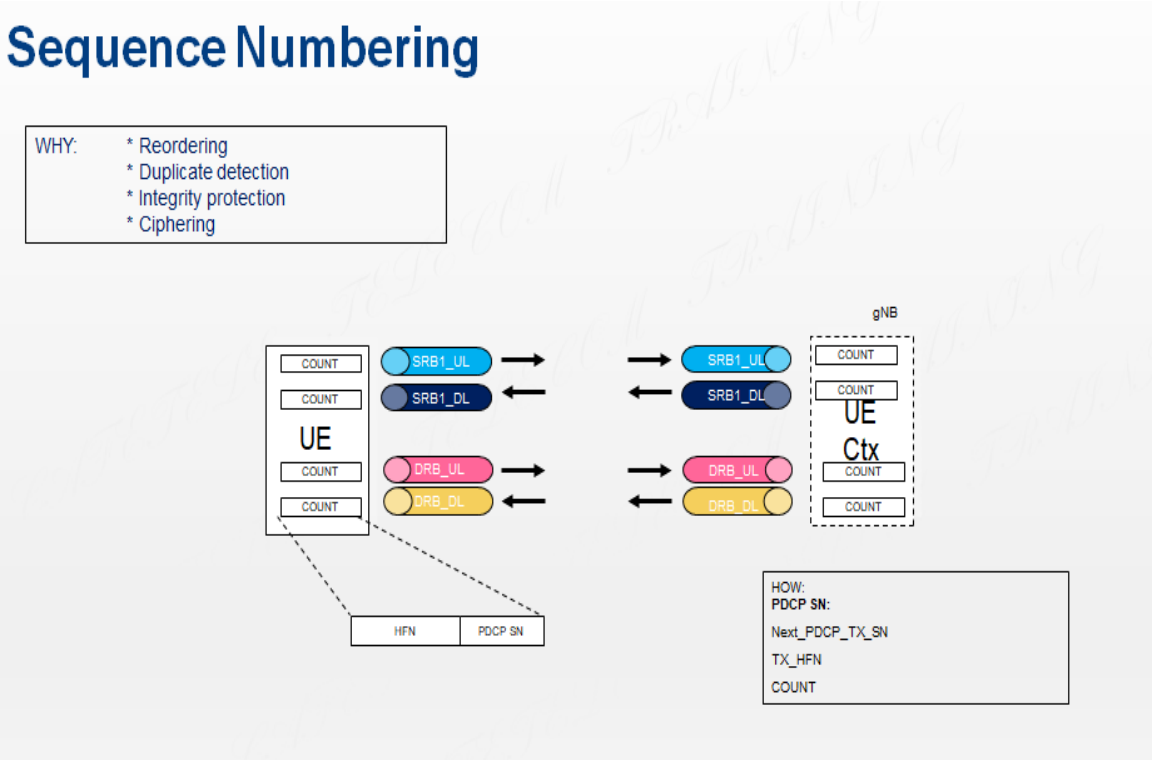
Sequence Numbering

In 5G networks, particularly in the Packet Data Convergence Protocol (PDCP) layer, sequence numbering plays a crucial role in ensuring reliable and ordered delivery of packets between the User Equipment (UE) and the Radio Access Network (RAN).



Purpose: Sequence numbering in PDCP is used to order and identify the packets sent between the UE and the RAN. It ensures that packets are delivered in the correct order and allows for the detection of lost or duplicated packets.

Sequence Number Field: Each PDCP protocol data unit (PDU) contains a sequence number field. This field typically ranges from 0 to 4095 (12 bits), allowing for up to 4096 unique sequence numbers before

wrapping around.

Incrementation: The sequence number is incremented for each PDU sent by the transmitter (UE or gNB). The receiver (gNB or UE, respectively) uses the sequence number to reorder received PDUs and detect any missing or duplicated packets.

Error Detection: Sequence numbers also help in error detection and recovery. If the receiver detects a missing sequence number (indicating a lost packet) or a duplicate sequence number (indicating a duplicated packet), it can take appropriate actions such as requesting retransmission or discarding duplicates.

Windowing: Sequence numbers in PDCP also facilitate window-based flow control mechanisms. They allow the transmitter to keep track of which PDUs have been acknowledged by the receiver, thereby controlling the amount of unacknowledged data in the network.

Wraparound: Since the sequence number field has a limited size (12 bits in most implementations), it wraps around after reaching its maximum value (4095). Wraparound handling mechanisms ensure that sequence numbers are correctly interpreted even after wraparound occurs.

Purpose->

Ordering of Packets: The primary purpose of sequence numbers is to maintain the correct order of packets transmitted between the User

Equipment (UE) and the Radio Access Network (RAN). Each packet sent by the transmitter (UE or gNB) is assigned a unique sequence number.

Error Detection and Recovery: Sequence numbers help in detecting lost or duplicated packets. The receiver (gNB or UE) uses the sequence numbers to identify missing packets (gaps in sequence numbers) or duplicates (repeated sequence numbers). This allows the receiver to request retransmission of lost packets or discard duplicates to maintain data integrity.

Flow Control: Sequence numbers support flow control mechanisms, enabling the transmitter to manage the amount of unacknowledged data in the network. This helps in optimizing data transmission and network efficiency.

Implementation Details:

Sequence Number Field:

In PDCP, each Protocol Data Unit (PDU) contains a sequence number field.

Typically, the sequence number field is 12 bits long, allowing for sequence numbers ranging from 0 to 4095.

The sequence number increments sequentially for each PDU sent.

Wraparound Handling:

Due to the finite size of the sequence number field (12 bits), it wraps around after reaching its maximum value (4095).

Wraparound handling mechanisms ensure that sequence numbers are

correctly interpreted even after wraparound occurs. This involves proper tracking and interpretation by both the transmitter and receiver.

Acknowledgment and Retransmission:

Upon receiving PDUs, the receiver sends acknowledgments (ACKs) indicating the highest sequence number received and successfully processed.

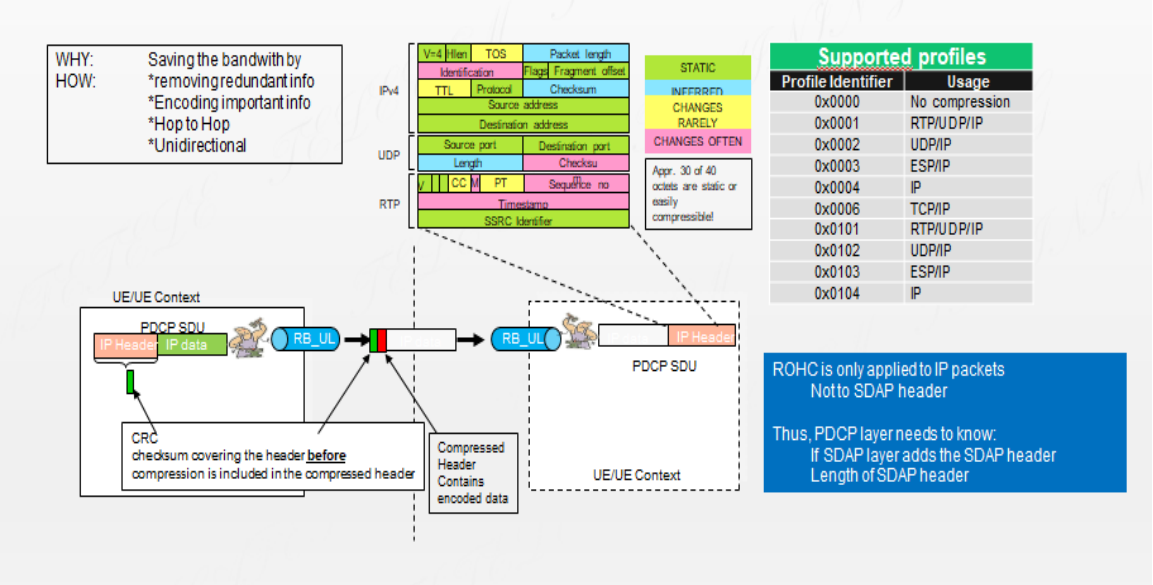
If the transmitter detects missing ACKs for certain sequence numbers (indicating lost PDUs), it retransmits those PDUs to ensure reliable delivery.

Duplication Detection:

The receiver uses sequence numbers to detect duplicate PDUs by recognizing repeated sequence numbers.

Duplicate PDUs can be discarded to prevent unnecessary processing and ensure data consistency.

Header Compression



Header compression in the Packet Data Convergence Protocol (PDCP) layer of 5G networks is an essential technique used to reduce the overhead associated with transmitting IP (Internet Protocol) packets over the radio interface.

Purpose of Header Compression:

Minimize Overhead: IP packets typically have headers that include various fields such as IP addresses, protocol information, and other control information. These headers can be relatively large compared to the payload data they carry. Header compression aims to reduce the size of these headers, thereby conserving valuable radio resources and increasing overall throughput.

Efficient Use of Radio Resources: By reducing the size of IP headers, header compression allows more efficient utilization of the limited

bandwidth available in the radio interface. This is particularly important in 5G networks where high data rates and low latency are required.

Improved Transmission Efficiency: Smaller headers mean less data to transmit, which can lead to faster transmission times and reduced latency for IP packets.

Techniques Used in PDCP for Header Compression:

ROHC (Robust Header Compression):

ROHC is a widely adopted standard for header compression in various communication protocols, including PDCP in 5G networks.

It operates by replacing repetitive or predictable fields in IP headers with shorter representations or by using context-based techniques to transmit only the changes between successive headers.

ROHC ensures that compression is robust against errors and packet loss, maintaining the integrity of compressed headers.

Context Management:

Header compression in PDCP involves maintaining context information between the transmitter (UE or gNB) and the receiver (gNB or UE).

Context information includes parameters necessary for compressing and decompressing headers, such as IP addresses, protocol identifiers, and sequence numbers.

Both ends of the communication must synchronize their contexts to

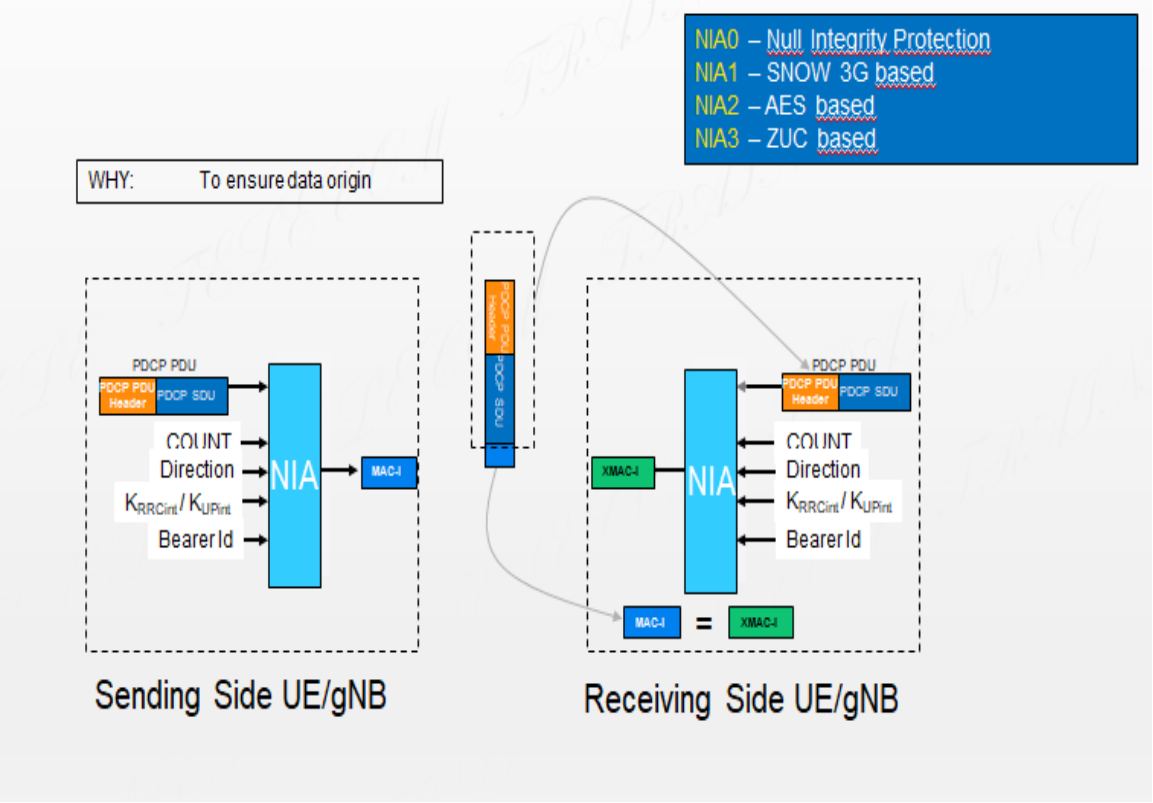
ensure that headers are correctly compressed and decompressed.

Dynamic Adjustment:

ROHC allows for dynamic adjustment of compression parameters based on the characteristics of the traffic and the radio conditions.

This adaptability ensures that header compression remains effective across varying network conditions, maximizing compression efficiency while minimizing overhead.

Integrity Protection



Integrity protection in the Packet Data Convergence Protocol (PDCP) is a crucial aspect of ensuring secure communication in mobile networks, specifically in 4G LTE and 5G systems. PDCP is a layer in the LTE/5G protocol stack responsible for header compression, security (encryption and integrity protection), and in-order delivery of data packets. Integrity protection ensures that the data has not been tampered with and guarantees the authenticity of the data by protecting it against unauthorized modifications.

Importance of Integrity Protection

Data Authenticity: Ensures that the data received is exactly what was sent by the sender, without any alterations.

Data Integrity: Protects data against tampering or modifications

during transmission.

Security: Plays a vital role in securing control and user plane data against potential attacks.

Working of Integrity Protection in PDCP:

Key Components

Integrity Algorithms: These algorithms compute integrity check values (ICVs) that are appended to the data packets.

Common algorithms include SNOW 3G, AES-CMAC, and ZUC.

Integrity Key (IK): A shared secret key used between the sender and receiver to generate and verify the integrity check values.

PDCP Control and User Plane: Both control (RRC messages) and user plane (data packets) data are protected by integrity algorithms.

Process of Integrity Protection

Key Generation and Distribution:

The integrity key (IK) is generated during the authentication and key agreement (AKA) procedure and securely distributed to the PDCP layer.

ICV Generation:

The PDCP layer uses the integrity key (IK) and an integrity algorithm to generate an integrity check value (ICV) for each packet.

The ICV is typically a fixed-length value (e.g., 32 bits) derived from the contents of the packet and the integrity key.

Appending ICV to Packet:

The ICV is appended to the packet before transmission. This ensures that any modification to the packet during transmission can be detected by the receiver.

Transmission:

The packet, along with its ICV, is transmitted to the receiver.

ICV Verification:

Upon receiving the packet, the receiver uses the same integrity key (IK) and algorithm to generate an expected ICV from the received packet's contents.

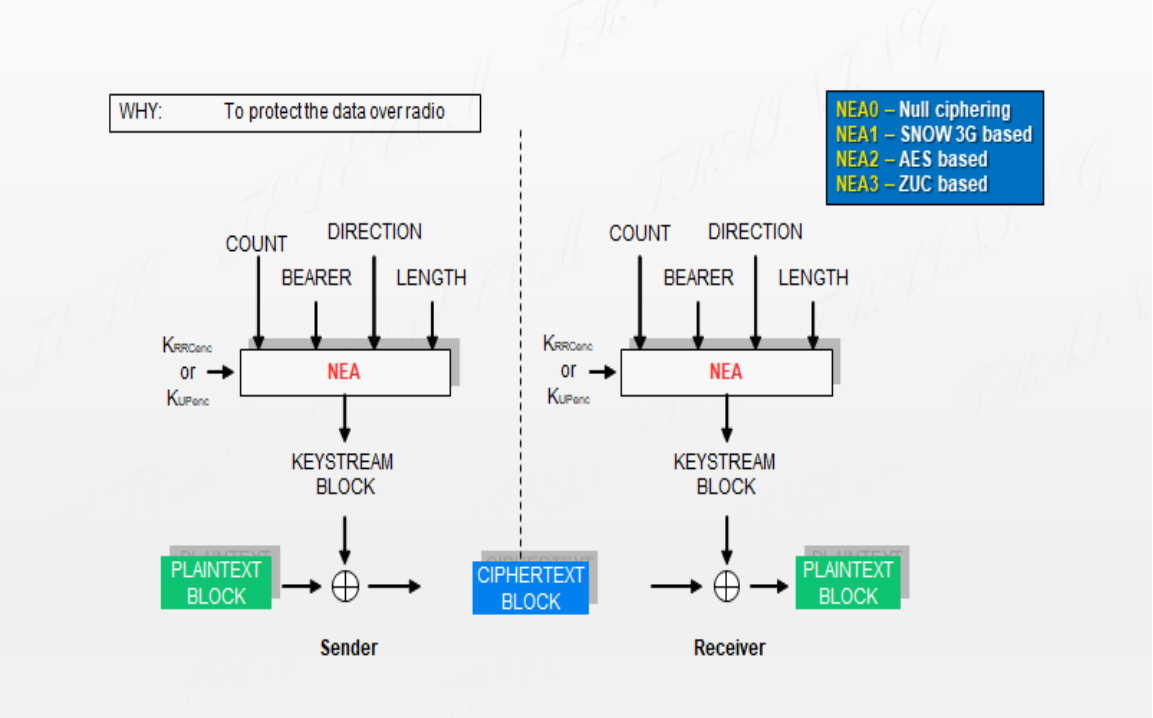
The receiver compares the computed ICV with the ICV appended to the packet.

Integrity Check:

If the ICVs match, the packet is considered authentic and intact.

If the ICVs do not match, it indicates that the packet has been tampered with, and the receiver discards the packet.

Ciphering



Ciphering, or encryption, in the Packet Data Convergence Protocol (PDCP) is a critical security feature used to protect the confidentiality of user data as it traverses mobile networks, including 4G LTE and 5G. The PDCP layer is responsible for several functions, including header compression, integrity protection, and ciphering. Ciphering ensures that the data remains confidential and is not readable by unauthorized entities.

Importance of Ciphering

Confidentiality: Ensures that user data cannot be intercepted and read by unauthorized parties.

Privacy: Protects the privacy of users by encrypting their data.

Security: Provides a secure communication channel between the user equipment (UE) and the network.

Working of Ciphering in PDCP

Key Components

Ciphering Algorithms: These algorithms are used to encrypt and decrypt the data. Common algorithms include:

SNOW 3G

AES (Advanced Encryption Standard)

ZUC (Zu Chongzhi algorithm)

Encryption Key (KeNB or K_gNB): A shared secret key used between the sender and receiver for encryption and decryption.

PDCP Control and User Plane: Both control plane (RRC messages) and user plane (data packets) data are protected by ciphering algorithms.

Process of Ciphering

Key Generation and Distribution:

The encryption key (KeNB or K_gNB) is derived during the

authentication and key agreement (AKA) procedure and securely distributed to the PDCP layer.

Ciphering of Data:

The PDCP layer uses the encryption key and ciphering algorithm to encrypt the data before transmission.

The encryption process transforms the plain text data into cipher text, which is unreadable without the correct decryption key.

Transmission:

The encrypted data (cipher text) is transmitted to the receiver over the air interface.

Deciphering of Data:

Upon receiving the encrypted data, the receiver uses the same encryption key and algorithm to decrypt the data.

The decryption process transforms the cipher text back into plain text, making it readable to the authorized receiver.